

**REMARKS****BEST AVAILABLE COPY****I. Introduction**

Claims 10 and 19 have been previously cancelled. Claims 1-2, 7-9, 11-12 and 16-18 have been amended. Thus, claims 1-9 and 11-18 remain pending in the present application. No new matter has been added. In view of the above amendments and the following remarks, it is respectfully submitted that all of the presently pending claims are allowable.

**II. The Claim Rejections Under 35 U.S.C. § 102(e) Should Be Withdrawn**

Claims 1-9 and 11-18 stand rejected under 35 U.S.C. § 102(e) as unpatentable over U.S. Patent No. 6,493,868 to DaSilva et al. ("DaSilva"). *01/26/06 Office Action*, p. 2.

DaSilva generally describes an integrated code development tool for developing and analyzing embedded real-time software. This code development tool enables a developer to configure, build, debug, trace, and analyze applications for digital signal processor (DSP) chips. (*See* DaSilva, col. 1, ll. 38-43). The integrated development tool includes DSP Code Generation Tools, Integrated Development Environment ("IDE"), and DSP/BIOS plug-ins and API calls. (*See* Id., col. 1, ll. 50-54). The DSP Code Generating Tools provides the environment for the development and production of DSP assembly language source code from compiled C source code. (*See* Id., col. 1, ll. 56-65). The IDE is designed to edit, build, and debug DSP target programs. (*See* Id., col. 2, ll. 25-28). The DSP/BIOS plug-ins and API calls provide support for real-time program analysis, allowing the developer to probe, trace, and monitor an application. (*See* Id., col. 2, ll. 62-67).

**BEST AVAILABLE COPY**

With respect to editing, the IDE allows the developer to “edit C and assembly source code” and to “view C source code with the equivalent assembly instructions shown after the C statements.” (*See Id.*, col. 2, ll. 30-33). The C source code with the equivalent assembly instructions shown after the C statements is viewable to the developer via an editor window. (*See Id.*, Fig. 4 and col. 1, ll. 23-25). Within the IDE, a Configuration Tool may be used to create files that define objects used by the DSP/BIOS plug-ins and API calls. (*See Id.*, col. 3, ll. 23-25). More specifically, the Configuration Tool sets the parameters used by the DSP/BIOS API calls at run-time, and it serves as a visual editor for creating and setting properties for run-time objects. (*See Id.*, col. 3, ll. 29-33).

The present invention relates to representation of data/file codes (or command nodes) using a command structure. Claim 1 recites a method executed on a computing device to perform an operation on extracted elements of a software code. Specifically, the software code is described to include a command node list, a parameter list, and a handler list. (*See Specification*, Fig. 12 and p. 25, ¶ 0043). In addition, the method of claim 1 includes the step the of “determining whether the extracted element is on the list of desired elements.” As described in the specification, once an element is extracted the extracted element is reviewed to determine if such element is one of the predetermined elements, wherein the predetermined elements may be defined by a user. (*See Id.*, Fig. 15 and p. 26, ¶ 0044). Accordingly, the present invention may allow a user to indicate which elements of a software code are to be extracted based on a review of a specific list of desired elements within the software code. (*See Id.*).

Claim 1 also recites the operations which are performed on the extracted element. Specifically, “the operation is one of generating a command data structure representation using

**BEST AVAILABLE COPY**

the command node element, generating handler function definitions and parameter definitions using the handler function element and the parameter element, and generating a handler function code using the handler function element and the parameter element. a command structure of the present invention may be generated using the command node element.”

According to the DaSilva description, Fig. 4 represents “an editor window for viewing C source code with the equivalent assembly instructions shown after the C statements.” (See DaSilva, col. 1, ll. 23-25). The editor window depicted in Fig. 4 simply provides a display of the C source code with the assembly source code (compiled C source code) to the developer. (See DaSilva, col. 2, ll. 31-33). A list of code may be displays through the editor window of DaSilva. However, the list in Fig. 4 does not describe the command list, the parameter list, nor the handler list. Furthermore, DaSilva does not describe an analogous method to performing the operations according to the amended claim 1 of the present invention. Thus, it is respectfully submitted that DaSilva’s disclosure neither teaches nor suggests a method including the step of “performing an operation on the extracted element when the extracted element is determined to be on the list of desired elements, wherein the operation is one of generating a command data structure representation using the command node element, generating handler function definitions and parameter definitions using the handler function element and the parameter element, and generating a handler function code using the handler function element and the parameter element” as recited in claim 1.

Applicants respectfully submit that for at least the reasons stated above, claim 1 of the present application is not anticipated by DaSilva, and request that the rejection of this claim be withdrawn. As claims 2-9 depend from, and therefore include all the limitations of claim 1, it

**BEST AVAILABLE COPY**

is hereby submitted that these claims are also allowable.

Claim 11 includes recitations substantially similar to those of claim 1. Such recitations include "determining whether the extracted element is on the list of desired elements" and "performing an operation on the extracted element when the extracted element is determined to be on the list of desired elements, wherein the operation is one of generating a command data structure representation using the command node element, generating handler function definitions and parameter definitions using the handler function element and the parameter element, and generating a handler function code using the handler function element and the parameter element." Therefore, Applicants respectfully submit that claim 11 is allowable for at least the reasons discussed above with regard to claim 1. Because claims 12-18 depend from, and therefore include all the limitations of claim 11, it is hereby submitted that these claims are also allowable.

**BEST AVAILABLE COPY****CONCLUSION**

In light of the foregoing, Applicants respectfully submit that all of the now pending claims are in condition for allowance. All issues raised by the Examiner having been addressed. An early and favorable action on the merits is earnestly solicited.

Respectfully submitted,

Dated: March 17, 2006

By: 

Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP  
150 Broadway, Suite 702  
New York, NY 10038